International Symposium & Exhibition on Geoinformation 2005, Penang, Malaysia, 27 – 29 September 2005

IMAGE DATABASE ON THE WWW WITH POSTGRESQL OBJECT-RELATIONAL DBMS: MEDICAL AND GIS APPLICATIONS

Deni Suwardhi[1], Halim Setan[1], Albert K. Chong[2] and Zulkepli Majid[1]

[1]Faculty of Geoinformation Science & Engineering, Universiti Teknologi Malaysia
[2]School of Surveying, University of Otago, Dunedin, New Zealand

Abstract

This paper presents the development of a web-driven database for images. The image database archives 2-D and 3-D images. Image files are stored unchanged in a central directory (archive). Users of the system are subdivided into groups, and a user who owns an image (by uploading it) can determine specific user on the system that is allowed to view and use the image. Files can be uploaded through HTML, or using a mounted (SAMBA, NFS, etc.) drive. Entering meta-data is facilitated by user-definable templates. The meta-data fields currently in use have been designed for medical images. The meta-data fields for image in GIS are also discussed. When combined with cheap, large hard drives and a file-save backup mechanism, this image database provides a perfect means to archive images within the setting of small to medium-sized research groups. Moreover, it might be useful wherever people collaborate on images. This image database consists of PHP scripts communicating with a POSTGRESQL database. It uses IMAGEMAGICK's convert to generate thumbnails from 2-D images, and EZDICOM library to generate thumbnails from 3-D images. The system has been successfully tested on WINDOWS XP Professional and LINUX (REDHAT 7.0), and should work with APACHE, PHP, POSTGRESQL and IMAGEMAGICK. Once these components are up and working, installation of this image database is simple with HTML installation form. Issues of data capture, storage, querying and retrieval in medical and GIS applications will be addressed as well.

Keywords: *image, medical, database, GIS, raster, blob, ORDBMS.*

1. INTRODUCTION

Image bases are an essential part of information systems and multimedia applications that brings their continuous development. The fact that images are much richer in image information than text and can be differently interpreted according to the application domain makes image data management very complex. There are variable and specific types of problems that appear in different applications and as a result the basic characteristics that must be identified and extracted from the images are very different.

Image databases (IDB) require large storage resources and usually a network information access. Due to this server technologies are used mainly. The continuous IDB development increases their capabilities and improves their basic characteristics: quickness, flexibility and the essential storage for the data. This imposes IDB design to integrate ideas and techniques of various areas of computer science such as computer graphics, image processing, image identification, artificial intelligence, database & knowledge base methodology, and Geographic Information System (GIS) of course. This composition of ideas culminates in new ideas for representation and new data models, exact and efficient algorithms for query processing and proper independent systems of architecture.

This paper presents the development of an image database as part of the research on craniofacial morphometric database. Large amounts of medical images of craniofacial organ are generated and collected with different imaging systems (e.g. CT, MRI, X-Ray, Photogrammetry, and Laser) for the research from the Universiti Sains Malaysia hospital in Kota Bharu. This leads to a growing need for a systematic storing and retrieving mechanism. Automated image analysis of these images is anticipated in order to provide important information, e.g. organ surface information or information

about abnormality and its patterns. Researchers are developing different algorithms for the automated image analysis. They need to have access to the images and later store their results with the images so the medical staff at the hospital can use them for their further analysis and judgments, statistical studies or to confirm that the detection is correct.

A web-based images archive application is proposed allowing joint access to the images by doctors and researchers even world wide and providing the required functions for storing, searching and accessing the images and their associated information. The first step of this development was to collect and evaluate the objectives and requirements of the desired web-based database application. Then the application had to be designed, which included deciding and choosing the appropriate database system, the database design, the programming language and a web server.

Finally the system had to be implemented and documented. Due to the first author's experience in GIS research and development, he tries to implement and to combine the IDB and GIS with PostgreSQL/PostGIS for web-based GIS data sharing (vector and raster) application. Satellite images and aerial images are the objects of this application.

2. STATE OF THE ART

This section deals with the state of the art in Image-Storing-And-Retrieval-Systems with regard to the field of medical, GIS and the Internet. After looking at different approaches in this section, the most suitable method to handle image for IDB is selected. A database with additional information is needed in order to be able to fulfill the wishes of researchers and surgeons for additional search criteria, e.g. to retrieve images based on an advanced combination of search criteria or by simply browsing.

The only wide-spread database system with special built-in features for images is DB2 from IBM with its DB2 Image Extender (www-306.ibm.com/software/data/db2). Features of this system are the import and the export of images and their attributes (such as size, format, height, distribution of colors) into and out of a database and the generation of thumbnails. It is possible to retrieve just the thumbnail or the original image as well as querying the attributes. Besides this there are some functions for image manipulation. The only drawback of this solution is the price, which is expensive.

Besides this, all other Database Management Systems (DBMS) do not have specific image handling capabilities. Consequently, three possibilities are left to handle an image with a DBMS. The first is to store the image in the file system and just store a reference in the DBMS, the second is to store the image as a BLOB (binary large object) and the last is to store the image in array data type. In this work, the focus of DBMS is PostgresSQL because it is an open source object-relational database management system (ORDBMS) pioneered many concepts that only became available in some commercial database systems much later.

2.1. Fileserver with Path

Traditionally, images data have been stored in sets of files. Storing the path in the database instead of the image itself offers the big advantage of browsing and using the image in other applications and contexts without any special handling. As the file system has no idea about the semantics (pixel type, number of dimensions, etc.), all selection and processing is burdened to the application developer, leading to tedious, repetitive work and an ill-defined consistency state of the data. A big disadvantage is that it results in lower security, but the advantage is that it is easier to implement.

2.2. Binary Large Object (BLOB)

Relational DBMSs, designed to scale well indeed, traditionally store multidimensional arrays as unstructured BLOBs ("binary large objects") introduced by Lorie as "long fields" (Lorie 1982). This

technique cannot give any support for operations beyond line-by-line access, something clearly not feasible for large archives. But, it is better for the consistency of the database approach to store the image as a BLOB than as a file with path. Files cannot be deleted without a database query. And if the database indicates that there is an image, it is really there. If only the path was stored, a change in the fileserver would make the database worthless until all the paths are corrected.

Furthermore, the database is an important advantage concerning security. It is very easy to perform a user authorization before giving any image to the client. If no access control was needed, the file system of the web server would be quite comfortable. The usage of BLOB will prevent the user from naming an image differently, manipulating it without authorization or storing it just somewhere and so losing the context to the original. Confusion or loss of data is therefore prevented.

PostgreSQL allows columns of a table to be defined as `bytea` data type to store binary data like images. The SQL standard defines a different binary string type, called `BLOB` or `BINARY LARGE OBJECT`. The input format is different from `bytea`, but the provided functions and operators are mostly the same. PostgreSQL has a *large object* facility, which provides stream-style access to user data that is stored in a special large-object structure. Streaming access is useful when working with data values that are too large to manipulate conveniently as a whole.

2.3. Array Data Type

Object-relational database systems (ORDBMSs) allow adding new data types, including access operations, to the server (Stonebraker, 1998). PostgreSQL allows columns of a table to be defined as variable-length multidimensional arrays. Arrays of any built-in or user-defined base type can be created. Arrays, however, are not a data type, but a data type constructor (“template”), parameterized with cell type and dimension. Such templates are not supported by ORDBMSs; hence a separate data type has to be defined for 2-D greyscale images, 2-D hyperspectral images, 3-D medical images, etc. Furthermore, server internal components are not prepared for the kind of operations occurring in client applications, therefore important optimization are difficult to implement. Hence, array data type is felt currently are not an option in image data management.

3. SYSTEM ARCHITECTURE

The CIDB’s (Craniofacial Image Database System) is a database system for groups working with 2-D and 3-D images and developed as a part of CMDB’s (Craniofacial Morphometric Database System). It lets researchers and doctors store and retrieve images, and lets they decide who else can view the images. CIDB’s not only stores the original file unchanged, but also data describing the image (henceforth called image meta-data), like a description and keywords. Templates make it easy to enter these image meta-data.

This Web-based CIDB’s medical image archive system consists of three major software modules: a client-tier, middle-tier, and a database management-tier. A database server with a hierarchical storage system comprises the database management-tier. An Apache Web server, with PHP scripting language, and a firewall reside in the middle-tier as gateway objects, controlling access to the database management-tier. Desktop computers with a Web client in the client-tier give users transparent interfaces to data objects stored in the database management-tier. The user interface was built using Javascript. Any World Wide Web browser that supports Javascript (e.g. Netscape or Internet Explorer) can be used to retrieve multimedia information stored in the CIDB’s.

3.1. Designing the System Architecture

The hardware and software components were selected and combined so that the system architecture provides the necessary image archive functions with the desired performance characteristics. As

illustrated in Figure 1, three-tier architecture has been designed to implement a flexible, scalable image archive system, using a low cost hardware and software products. This design includes a secure interface to support database query as well as image and textual data storage and retrieval functions. System security is essential for the protection of medical image, patient information, clinical and research data. The server-based, multi-tiered model provides a responsive, reliable and secure architecture for distributed transactional applications.

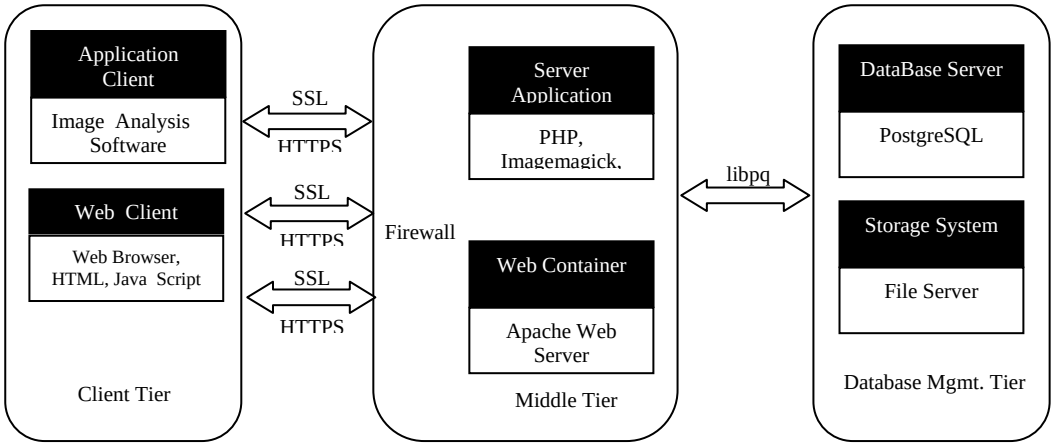


Figure 1. CIDB's Architecture

The database management tier is implemented with a database server acting as a front-end to the online and near-line archive storage system where the image data is located. The archive storage is organized in a two-level hierarchy. The meta-data and frequently accessed images were stored in the first level of the hierarchy, where recently images are stored as a large object file. Older images were stored in storage system units that form the second level of the hierarchy. The image database is also organized in a hierarchical structure and is indexed to facilitate the rapid searching for information.

The database hierarchy progresses from complete clinical studies to the patients within each study to the raw and processed image data. The database server interacts with textual data and metadata abstractions to index the image database that is used for accessing image data files stored in the two levels of the storage hierarchy. The textual data and metadata abstractions refer to descriptive information or attribute about the image data, and are used in the form of keywords for the query and retrieval of the image data. Indexes are usually constructed manually by analyzing textual data and visual attributes of related images to find and organize important concepts for textual and visual queries.

The middle tier, the application server layer, consists of data rules to assist in resource sharing. Data rules are executed on the database server to manipulate the database. It manages the entire workflow process from data acquisition to data storage. It automates the loading, archiving, and subsequent display and delivery of medical images and metadata. Using an open source Apache web server and PHP script as the middleware platform, clients can access the Web server through Web browsers using HTML. The web server will access the image database using *libpq* library. Clients can also access the server with stand-alone applications, such as image viewer and analysis programs.

The client tier provides a graphical user interface (GUI) and interactive software to grant the user access to the image archive. It presents users with patient data and related images for viewing, manipulation, and data entry. The two main types of user interface for this layer are the traditional standalone application and the Web-based application.

3.2. System Components

In order to store, retrieve, and visualize the images data, a diverse set of tools, components and products that are both Open Source and our Developed Software packages were used. The following integrated toolsets are used:

- Windows/PostgreSQL 8.0.2 is implemented on a PC Server, which contains one Intel Pentium 4 2.4 MHz processors, 1 GB of memory.
- Apache Web Server version 2.0.50.
- PHP version 4.3.7
- It uses Imagemagick's convert to generate thumbnails from 2-D images, and ezDICOM library to generate thumbnails from 3-D images.
- Finally, end user queries and data extraction operations are performed with Web browser such as Firefox, Netscape, or Internet Explorer depending on user requirements

The system components described above are connected to a 100MBps Ethernet network. This network connectivity allows a typical request for 50 Mbytes of image data to be transmitted in approximately five seconds or a larger request for 500 Mbytes of image data to be transmitted in approximately one minute, including overhead in both examples.

3.3. System Security

To secure the image archive system from unauthorized access, the web server will use Apache-SSL web server to ensure the confirmation of the user's identity by validating the certificate and public ID. Apache-SSL is a secure Webserver, based on Apache and SSLeay / OpenSSL. It is licensed under a BSD-style licence, which means, in short, that users are free to use it for commercial or non-commercial purposes (http://www.apache-ssl.org/). An encryption algorithm was also implemented, which encrypts patient information and user passwords, and a user login locking mechanism, which allows only three failed login attempts, thus providing a high degree of confidentiality for sensitive data and prevention of unauthorized system access (Figure 2).

A two-layer access-authentication management scheme was implemented to give different users different access level privileges to the image archive database. This authentication management scheme controls all access and operations at the system and database levels, down to the individual data files stored in the archive database.

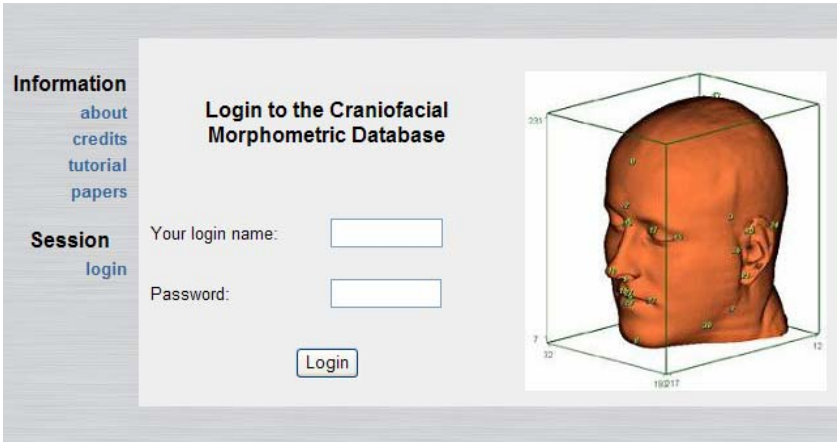


Figure 2. Window's Login for security

4. DATABASE DEVELOPMENT

The image database system is designed to store large binary image objects linked with clinical data and associated metadata. It supports queries across multiple imaging studies. An imaging study refers to a set of CT, X-Ray, MRI, Photogrammetry and laser images collected for a single purpose on a single patient. Querying across collections of these studies allows for searching and retrieving multiple image data sets for common features. An example query would be to find the CT images of all male patients within a certain range of ages who have been treated with a specific medical procedure.

4.1. Database Design

The information stored in the image archive system was divided into textual data and binary image data. The textual data, which is subject to indexing and querying, consists of metadata, patient data, and clinical data, such as examination technique, diagnosis, abnormality classification and treatment. The binary image data consists of CT, X-Ray, MRI, Photogrammetry and Laser images. The clinical data comprises over 50 attributes organized into five groups as follows: Patient Information, Studies, Exams, Diagnosis, and Treatment.

The textual data and the binary image data were primarily organized by patient. Each patient is assigned a unique subject ID number to avoid a situation where a patient is assigned multiple patient record numbers. This subject ID number is also used in database tables as a foreign key to ensure a deletion of all related data from the database when a patient record is deleted from the database. The patient and clinical data, including metadata, are stored in the first level of the hierarchy of the archive storage to ensure rapid access. The image metadata includes the results derived from the original patient images.

Figure 3 shows an entity-relationship (E-R) diagram capturing a subset of the full schema for this image archive system.

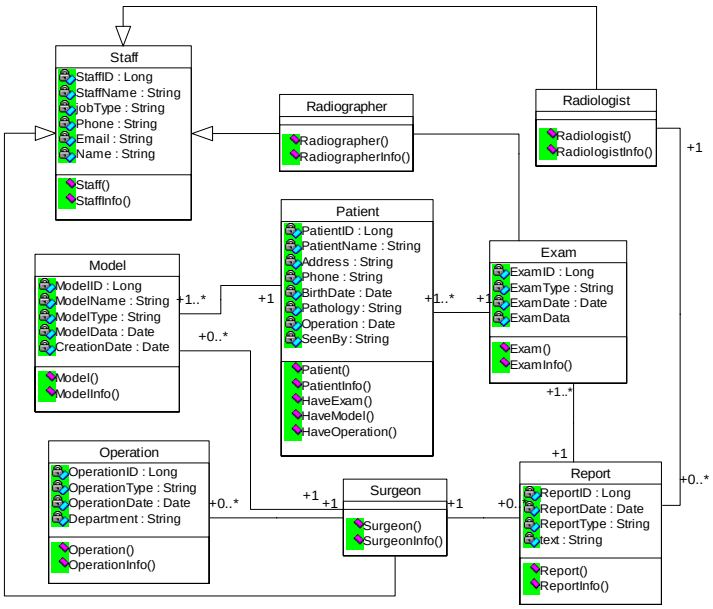
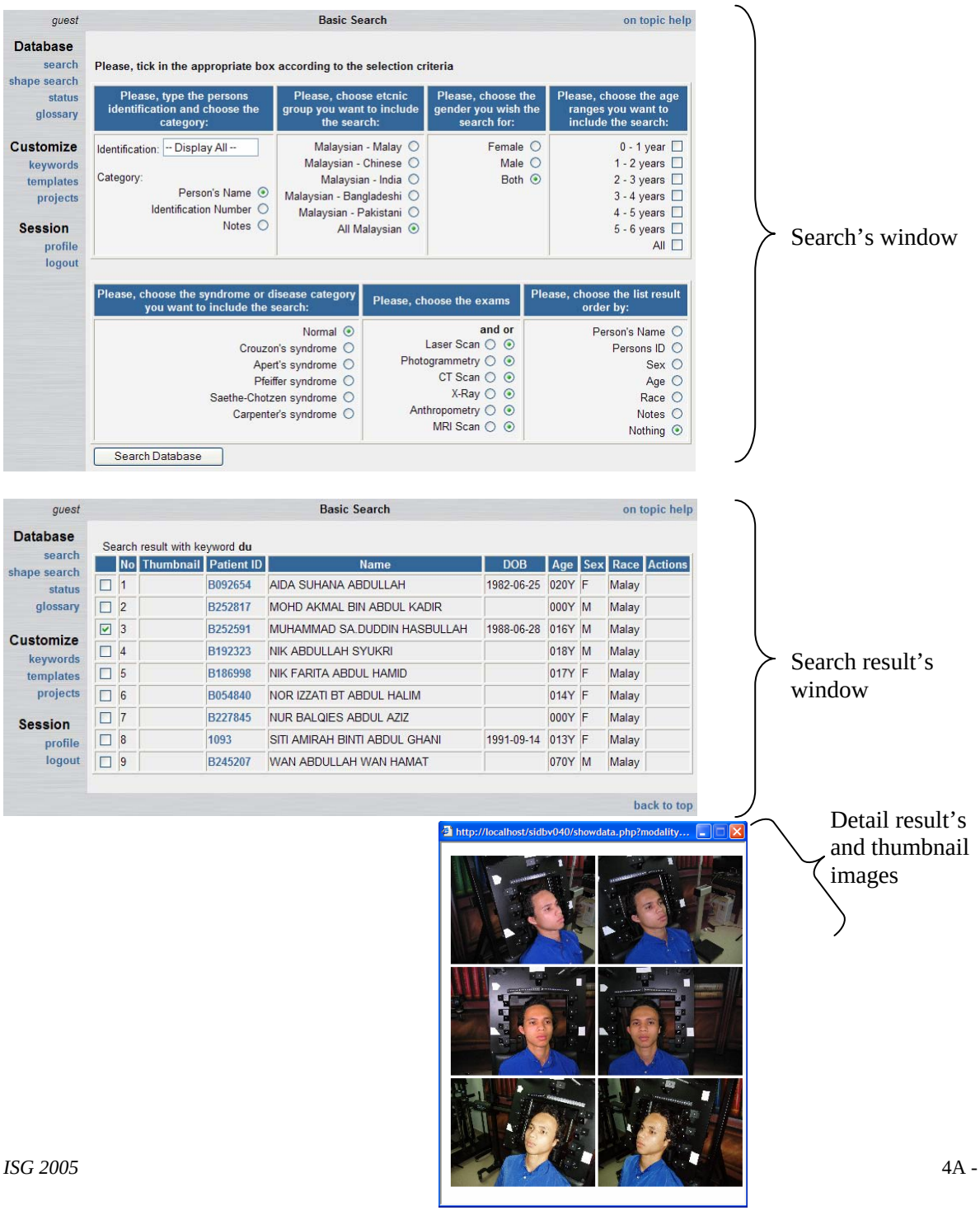


Figure 3. A Subset of Entity-Relationship Diagram of the CIDB's Schema

4.2. Database and Query Implementation

The implementation stage is a physical realization of the database design as described in the previous section. This implementation is achieved using the Data Definition Language (DDL) to create database schema and empty database files. The Data Manipulation Language (DML) is also to facilitate database transactions, either update transaction or retrieval transaction. In this work, either DDL or DML is precompiled in PHP scripting language and implemented via libpq library.

Using provided Web interfaces, users can build a customized database query using many selectable query criteria. The Web browser passes the customized database query to the PHP session in the web server, which then sends the query to the database server and obtains the query results for presentation to the user via the Web. All the queries are targeted to retrieve the medical images that meet the query criteria. A query tool, as shown in Figure 4 is provided for users to download image data files as well as patient information and clinical data. The download tool allows user to navigate the local computer hard drives to select a directory where the downloaded data can be saved.



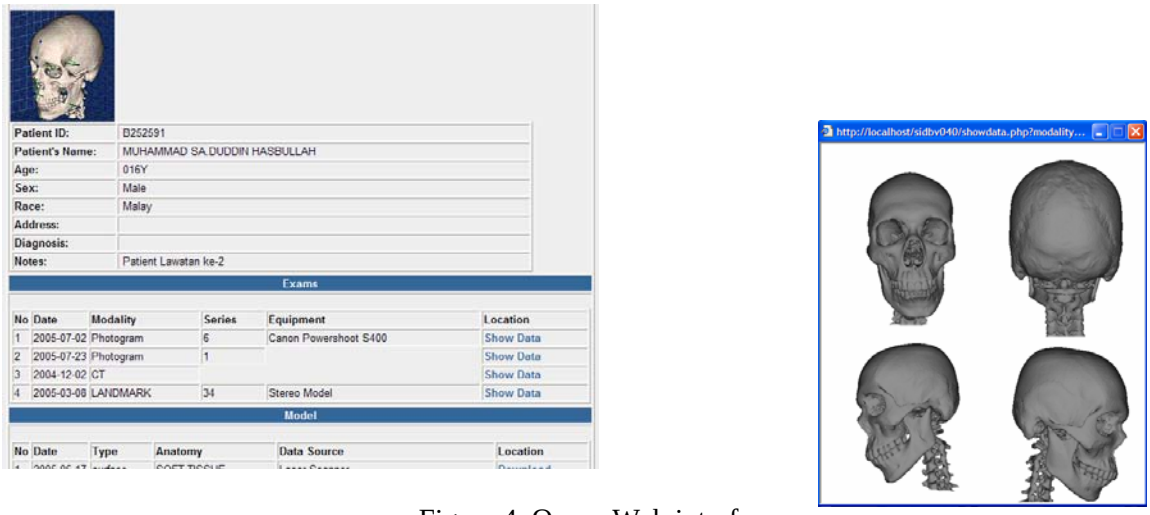


Figure 4. Query Web interfaces

As shown in Figure 5, the user administration Web page gives the system administrator a user-friendly interface to add, delete, activate, and inactivate a user, as well as set the access privilege level. After a user requests a registration through a User Registration Web page, the system administrator can then activate the user and set an access privilege level for accessing the image archive system. The access privilege level is largely categorized as follows: first, different level of database access functionality (e.g., upload, query, and download); and second, limits on system visits (e.g., access counts and days).

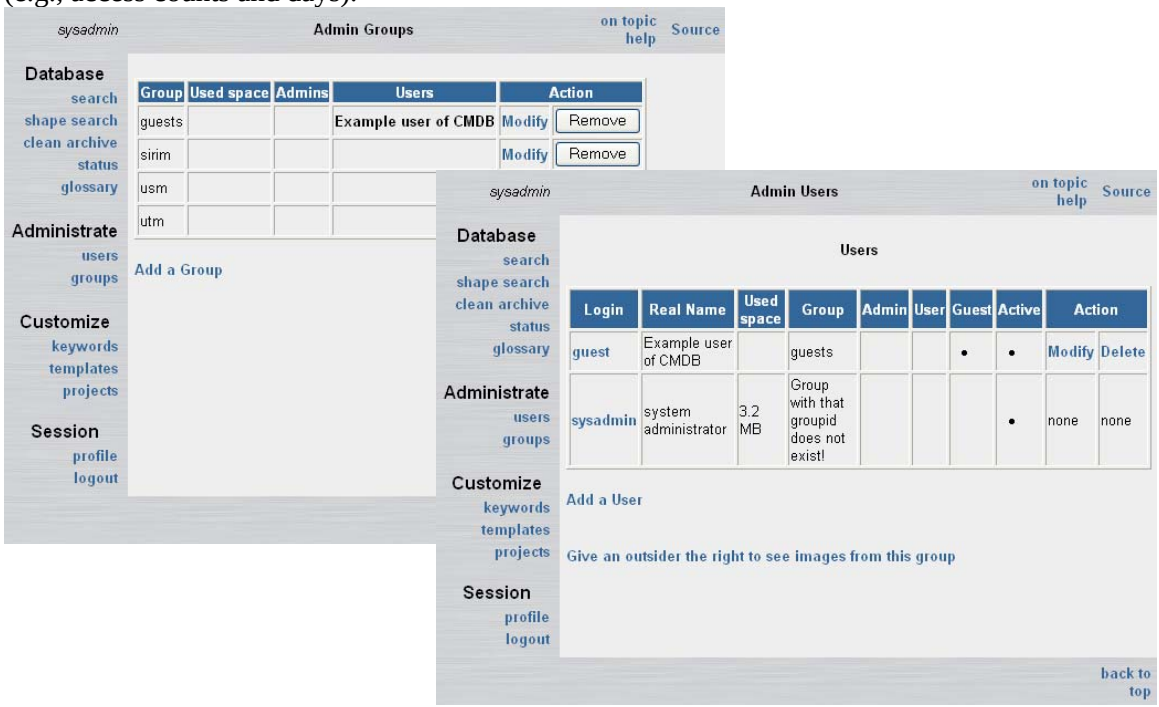


Figure 5. Administration Web Pages

4.3. Future Development

To provide users with the capability to search and retrieve images using content information (e.g., size, shape and color), the authors plan to develop and implement the capability of content-based visual information retrieval. Images can be stored in the database as the BLOB (Binary Large Object) data type, which can be used to perform analysis, data mining, and content-based visual information retrieval functions (Zhu *et.al.*, 1996). To allow the visual selection of images by criteria that cannot be expressed in textual form (Stahl and Kramann, 1997), the authors also plan to implement the

capability to thumbnail preview images stored in the archive database. When an image is loaded into the database, the system can generate a thumbnail of the image that can then be stored with the original image. Finally, as a sub-component to the image archive database, the authors plan to develop and implement a distributed database scheme that allows geographically distributed database systems located at other research sites and containing additional research data to be connected to the central archive system over the Internet (Abdelguerfi and Wong, 1998).

5. IMPLEMENTATION CIDB'S FOR GIS

In the area of geographical information systems (GIS), database technology has enabled the complete integration of spatial data types in object-relational data base management systems, creating a new generation of spatial databases (Shekhar, Chawla et al. 1999). This integration is bound to change completely the development of GIS technology, enabling a transition from the monolithic systems of today (that contain hundreds of functions) to a generation of spatial information appliances, small systems tailored to specific user needs (Egenhofer, 1999). Three major challenges in spatial database construction are: (a) the efficient handling of spatial data types, which include both vector (i.e., polygons, lines and points) and raster data structures; (b) the availability of tools for query and manipulation of spatial data; (c) the support for advanced applications, such as mobile GIS, spatiotemporal data models (Hornsby and Egenhofer 2000), and geographical ontologies (Fonseca, Egenhofer et al. 2002).

One area of special interest is the efficient handling of raster data, especially satellite images and aerial images. Remotely sensed imagery is one of the vastly used spatial data currently available to geographic researchers. The variety of spatial and spectral resolutions for remote sensing images is large, ranging from IKONOS 1-meter panchromatic images to the radar images. Recent advances in remote sensing technology, with the deployment of a new generation of sensors, have improved considerably such application areas as environmental monitoring and urban management.

In order to store the raster data in a DBMS, previous works in the literature (Patel, Yu et al. 1997; Reiner, Hahn et al. 2002) have shown than a combination of *multi-resolution pyramid* and a *tiling* scheme is the most appropriate strategy for handling large image files. The tiling scheme is used as a spatial index. When retrieving a section of an image, only the relevant tiles will be retrieved and decompressed. The multi-resolution pyramid is very useful for visualization of large data sets, to avoid unnecessary data access. This approach will be adopted in CIDB's for GIS. Each image is associated to a database table, called the "raster table", where each record stores a tile (or block) of the image. The fields of the raster table are shown in Table 1.

Table 1. Raster Table

Block_id	`String`	Tile unique identifier
Lower_x	`real`	Tile minimum x coordinate
Lower_y	`Real`	Tile minimum x coordinate
Upper_x	`Real`	Tile minimum x coordinate
Upper_y	`Real`	Tile minimum x coordinate
Band_id	`Int`	Band
Resolution_factor	`Int`	Tile resolution factor
Subband	`Int`	Subband information
Spatial_data	`Bytea/Blob`	Raster tile
Block_size	`int`	Size in bytes of a tile (after compression)

For image storage in a CIDB's for GIS, each band of the image is decomposed in a set of disjoint tiles with width W and height H given in number of pixels. Each tile is stored in the field spatial data (as a BLOB) of a record in a raster table. Improvements can be made by using an r-tree data structure in PostgreSQL to index the tiles. Tiles can be compressed before storage. Global information about a

raster layer, such as tile dimension, compression technique, number of rows and columns, number of bands, bounding box, and pixel resolution is stored in a complementary table.

To make integration between GIS and CIDB's and to implement multi-resolution pyramid and a tiling schema, Postgresql with PostGIS is used for spatial (vector and raster) data management. PostGIS adds support for geographic "vector" objects to the PostgreSQL object-relational database. In effect, PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS), much like ESRI's SDE or Oracle's Spatial extension. PostGIS follows the OpenGIS "Simple Features Specification for SQL".

6. CONCLUSION

This paper presents a Web-based medical image archive system that has a three-tier, client-server architecture. The three-tier structure that has a database management-tier, a middle-tier, and a client-tier was described. This compartmentalized three-tier design allows us to accommodate an increase in the number of users and the amount required storage, upgrade the various components of the system, and implement additional systems for different research environments. The database design accommodates large binary object files external to the database. Though the image database is designed to meet the specific craniofacial morphometric database research needs, we feel that our design could easily be used for other image archive applications like GIS application. We described our implementation of user-friendly, configurable Web-interfaces, which provide users with easy access to the image archive system from their desktop computers over the Internet. The user administration Web page allows the system administrator the ability to control user level privileges for uploading data files, querying databases, and downloading data files remotely. We also developed tools for image uploading and downloading.

ACKNOWLEDGMENT

This work is part of a prioritized research IRPA vote 74537 sponsored by Ministry of Science, Technology & Innovation Environment (MOSTI) Malaysia.

REFERENCES:

1. Abdelguerfi, M. and Wong, K.F. (1998). *Parallel database Techniques*, IEEE Computer Society.
2. Egenhofer, M. (1999). *Spatial Information Appliances: A Next Generation of Geographic Information Systems*. First Brazilian Workshop on GeoInformatics, Campinas, Brazil.
3. Fonseca, F., M. Egenhofer, et al. (2002). *Using Ontologies for Integrated Geographic Information Systems*. Transactions in GIS 6(3): 231-257.
4. Hornsby, K. and M. Egenhofer (2000). *Identity-Based Change: A Foundation for Spatio-Temporal Knowledge Representation*. International Journal of Geographical Information Science 14(3): 207-224.
5. Lorie, R.A.(1982). Issues in Databases for Design Transactions. in: Encarnaçao, J.; Krause, F.L. (eds.): *File Structures and Databases for CAD*, North-Holland Publishing.
6. Patel, J., J. Yu, et al. (1997). *Building a Scalable Geo-Spatial DBMS: Technology, Implementation, and Evaluation*. SIGMOD Conference, Tucson, Arizona.
7. Reiner, B., K. Hahn, et al. (2002). *Hierarchical Storage Support and Management for Large-Scale Multidimensional Array Database Management Systems*. 3th International Conference on Database and Expert Systems Applications (DEXA), Aix en Provence, France.

8. Shekhar, S., S. Chawla, et al. (1999). *Spatial Databases: Accomplishments and Research Needs*. IEEE Transactions on Knowledge and Data Engineering 11(1): 45-55.

9. Stahl, J.N. and Kramann, B.(1997). Customized Medical Image Databases: A Low-Cost Approach. *Computerized Medical Imaging and Graphics*, 21(6): 345-350.

10. Stonebraker, M., and Brown, P. (1998). *Object-Relational DBMSs, Tracking the Next Great Wave*. Morgan Kaufman Publishers, Inc.

11. Zhu, X., Lee, K., Levin, D.L., Wong, S.T.C, Huang, H.K., Hoo, K.S., Gamsu, G. and Webb, W.R. (1996). Temporal Image Database Design for Outcome Analysis of Lung Nodule. *Computerized Medical Imaging and Graphics*, 20(4): 347-356: 1996.